SOME APPLICATIONS OF MAPLE IN LINEAR SYSTEMS ANALYSIS

J. Jones [1], N. P. Karampetakis [2] and A. C. Pugh [1]

INTRODUCTION

Many problems in linear systems analysis, as is true in many areas of mathematics, are seen to be very *computationally attractive*. By this we mean that it is in general beneficial to implement the corresponding problem computationally rather than solving the problem by hand in terms of certain governing factors. Such factors may include the time, accuracy and cost of obtaining such a solution.

The advent of complete symbolic computational packages, such as Maple [2] and Mathematica [9], have made the computational implication of such problems particularly attractive. Their main advantages, in regards to linear systems, lie in the following.

Symbolic Storage – Variables can be stored in an "exact" form (i.e. 1/3 as opposed to 0.3333....) resulting in no loss of accuracy during a calculation. Additionally variables can be left "unassigned" (i.e. without holding any numerical values) which enables polynomial operations to be defined, say, in an indeterminate s.

Inbuilt Procedures – The existence of hundreds of inbuilt procedures covering both general and certain specialized areas of mathematics. In Maple, for example, the package `linalg` consists of numerous procedures used in the solution of problems in linear algebra such as the inverse of a non–singular matrix and the solution of a set of linear equations.

Programming Language – The existence of unique high–level programming languages allowing specific procedures to be written. In Maple over 90% of these inbuilt procedures are written in its own procedural programming language and therefore any of them can be implemented in any new procedures written. They can therefore be viewed upon as *building blocks* for subsequently more specified and advanced procedures.

Perhaps understandably, however, neither one of these symbolic computer packages contain existing procedures or indeed packages for specialized work in linear systems. However they are clearly flexible enough for the implementation of such work to be carried out. This is the motivation of this paper and we present a package of procedures, developed for use in Maple [2], for solving numerous and common problems in linear systems.

THE LINEAR SYSTEMS PACKAGE

The linear systems package is split into three main topic areas namely those of *System Structures, System Representations* and *System Solutions*. These can each be looked upon as separate sub–packages themselves and are loaded into the Maple session independently.

1. System Structures (Maple Package: `linstruct`) – Contains a package of procedures concerned with determining the structure and properties of a system. It also includes several matrix operation procedures which can be seen as extensions to the existing linear algebra package `linalg`. The package currently consists of 25 procedures. Several of these procedures are discussed below:

`smithmillan(A,p,m,'l','r')` – Computes the Smith–McMillan form [6] $S^{\mathbb{C}}_{T(s)}$ of a rational matrix $T(s) \in \mathbb{R}(s)^{p\times m}$ where

$$T(s) = l(s) \times S^{\mathbb{C}}_{T(s)} \times r(s) \tag{1}$$

[1]Department of Mathematical Sciences, Loughborough University, Loughborough, Leics LE11 3TU. ENGLAND
[2]Department of Mathematics, Aristotle University of Thessaloniki, 54006. GREECE

Optionally gives the corresponding unimodular matrices $l(s) \in \mathbb{R}[s]^{p\times p}$ and $r(s) \in \mathbb{R}[s]^{m\times m}$ as in (1).

`rowprop(T,'U')` – Performs unimodular row operations on a matrix $T(s) \in \mathbb{R}[s]^{p\times m}$ so as to form a matrix $T_{prop}(s) \in \mathbb{R}[s]^{p\times m}$ which is row proper. i.e. its high order row coefficient matrix has full rank. Optionally computes the unimodular matrix $U(s) \in \mathbb{R}[s]^{p\times p}$ where

$$T_{prop}(s) = U \times T(s) \tag{2}$$

The procedure uses the algorithm as seen in [8] and this has been extended for the column proper case for use in the analogous procedure `colprop(T,'U')`.

`MINrat(A,q,'U')` – Computes the greatest degree of all $(q \times q)$ minors of a matrix $A(s) \in \mathbb{R}(s)^{p\times m}$. Optionally also computes and stores all these associated $(q \times q)$ minors.

`SmRatinf(G)` – Computes the Smith–McMillan form $S^{\infty}_{G(s)}$ of a matrix $G(s) \in \mathbb{R}(s)^{p\times m}$ at $s = \infty$ as defined in [8]. The corresponding algorithm utilizes the procedure `MINrat` above in that the high–order $(q \times q)$ minors, where $q = 1, \ldots, min(p, m)$, are required.

`ginverse(G,p,m)` – Computes the generalized inverse $G(s)^{\dagger} \in \mathbb{R}^{m\times p}$ of a matrix $G(s) \in \mathbb{R}(s)^{p\times m}$, as defined in [5], where $G(s)$ is non–square or is square and singular. The procedure uses the algorithm of [3] which holds for when $G(s)$ is of constant, polynomial or rational form.

`Forlaur(A,q)` – Computes the Forward Fundamental Sequence of a matrix $A(s) \in \mathbb{R}[s]^{r\times r}$ [4]. This is defined as the sequence of matrices H_j, $j \leq v$ where

$$A(s)^{-1} = H_{\hat{q}_r} s^{\hat{q}_r} + H_{\hat{q}_r - 1} s^{\hat{q}_r - 1} + \cdots + H_1 s + H_0 + H_{-1} s^{-1} + \cdots \tag{3}$$

is the Laurent expansion of $A(s)^{-1}$ about the point $s = \infty$. This sequence is of great importance in the solution of ARMA–Representations [4]. This also applies to the corresponding Backward Fundamental Sequence and a corresponding procedure `Backlaur(A,q)` is therefore included.

2. System Representations (Maple Package: `linrep`) – Contains a package of procedures used to determine alternative and equivalent representations of a system. It uses several procedures from the package `linstruct` above and these are therefore included. (NOTE: The procedures `BOSred`, `PenEquiv`,`linindcols` and `linindrows` also require the existing Maple procedure `choose`). The package currently consists of 25 procedures

The procedures contained within the package `linrep` can be sub–divided into the following areas and several procedures are discussed:

Right & Left MFD's Procedures – Procedures for representing a matrix $G(s) \in \mathbb{R}(s)^{p\times m}$ as a right (*resp.* left) Matrix Fraction Description

$$G(s) = N_1(s) D_1(s)^{-1} \quad \left(= D_2(s)^{-1} N_2(s)\right) \tag{4}$$

where $N_1(s) \in \mathbb{R}[s]^{p\times m}, D_1(s) \in \mathbb{R}[s]^{m\times m}$ (*resp.* $D_2(s) \in \mathbb{R}[s]^{p\times p}, N_2(s) \in \mathbb{R}[s]^{p\times m}$).

`LMFD(G,p,m)` – Computes a left MFD of a matrix $G(s) \in \mathbb{R}(s)^{p\times m}$ where the corresponding matrices $D_2(s) \in \mathbb{R}[s]^{p\times p}$ and $N_2(s) \in \mathbb{R}[s]^{p\times m}$ are left coprime. i.e.

$$S^{\mathbb{C}}\left(D_2(s)\ N_2(s)\right) = \left(I_p\ 0_{p,m}\right) \tag{5}$$

where $S^{\mathbb{C}}(\cdot)$ denotes the Smith–McMillan form of the indicated matrix. Similarly the procedure `RMFD(G,p,m)` computes a right coprime MFD.

`LMFDrowred(G,p,m)` – Computes a left MFD of a matrix $G(s) \in \mathbb{R}(s)^{p\times m}$ where the corresponding matrices $D_2(s) \in \mathbb{R}[s]^{p\times p}$ and $N_2(s) \in \mathbb{R}[s]^{p\times m}$ are left coprime (see above) and the the matrix $(D_2(s)\ N_2(s))$ is row proper (see above). This type of MFD has direct implications in realization theory. Similarly the procedure `RMFDcolred(G,p,m)` computes a right coprime MFD of a matrix $G(s)$ where the corresponding matrix $\begin{pmatrix} D_1(s) \\ N_1(s) \end{pmatrix}$ is column reduced where $D_1(s)$ and $N_1(s)$ are as given in (4).

Realization Procedures – Procedures for realizing polynomial and strictly proper matrices $G_{pol}(s) \in \mathbb{R}[s]^{p\times m}$ and $G_{sp}(s) \in \mathbb{R}(s)^{p\times m}$ respectively.

`realprop(G,p,m)` – Given a strictly proper matrix $G(s) \in \mathbb{R}(s)^{p\times m}$ this computes matrices $J \in \mathbb{R}^{n\times n}$, $B \in \mathbb{R}^{n\times m}$ and $C \in \mathbb{R}^{p\times n}$, where J is in block Jordan form, such that

$$G(s) = C(sI - J)^{-1}B \tag{6}$$

This is based on an algorithm as seen in [8] and $[J, B, C]$ is a *minimal* realization of $G(s)$ i.e. the dimension n is minimal.

`realpoly(G,p,m)` – Given a polynomial matrix $G(s) \in \mathbb{R}[s]^{p\times m}$ this computes matrices $J_\infty \in \mathbb{R}^{\mu\times\mu}$, $B_\infty \in \mathbb{R}^{\mu\times m}$ and $C_\infty \in \mathbb{R}^{p\times\mu}$, where J_∞ is in block Jordan form, such that

$$G(s) = C_\infty(I - sJ_\infty)^{-1}B_\infty \tag{7}$$

Similarly $[J_\infty, B_\infty, C_\infty]$ is a *minimal* realization of $G(s)$ i.e. the dimension μ is minimal.

`MINrealpoly(G,p,m)` – Uses the procedure `realpoly` to form a minimal realization $[J_\infty, B_\infty, C_\infty]$ of a polynomial matrix $G(s) \in \mathbb{R}[s]^{p\times m}$. Given this realization the matrices $\hat{J}_\infty, \hat{B}_\infty, \hat{C}_\infty$ and $\hat{D}_\infty$ are computed such that

$$G(s) = \hat{C}_\infty(I - s\hat{J}_\infty)^{-1}\hat{B}_\infty + \hat{D}_\infty \tag{8}$$

where, in general, $\hat{J}_\infty$ has lower dimension than that of J_∞

Reduction Procedures – Procedures for reducing a Polynomial Matrix Description (PMD) to an equivalent Generalized State–Space (GSS) form. i.e. given a PMD [6]

$$\begin{aligned} T(\rho)\beta(t) &= U(\rho)u(t) \\ y(t) &= V(\rho)\beta(t) + W(\rho)u(t) \end{aligned} \tag{9}$$

where $\rho = \frac{d}{dt}, T(\rho) \in \mathbb{R}[\rho]^{r\times r}$ $(|T(\rho)| \neq 0)$, $U(\rho) \in \mathbb{R}[\rho]^{r\times m}, V(\rho) \in \mathbb{R}[\rho]^{p\times r}$ and $W(\rho) \in \mathbb{R}[\rho]^{p\times m}$ compute a *minimal* positive integer $\sigma \in \mathbb{Z}^+$ and matrices $E \in \mathbb{R}^{\sigma\times\sigma}, A \in \mathbb{R}^{\sigma\times\sigma}, B \in \mathbb{R}^{\sigma\times m}, C \in \mathbb{R}^{p\times\sigma}$ and $D \in \mathbb{R}^{p\times m}$ such that the GSS model described by

$$\begin{aligned} E\rho x(t) &= Ax(t) + Bu(t) \\ y(t) &= Cx(t) + Du(t) \end{aligned} \tag{10}$$

possesses the same finite and infinite behavior as the corresponding PMD model.

`BOSred(T,r,p,m,'U','V')` – Uses the reduction algorithm of Bosgra & Van Der Weiden [1].

`VRED(G,r,p,m)` – Uses the reduction algorithm of Vardulakis [7]. This algorithm, in general, is superior to that of Bosgra & Van Der Weiden in that it computes a GSS model with lower dimension σ.

3. System Solutions (Maple Package: `linsol`) – Contains a package of procedures concerned with obtaining solutions to a system. It is primarily concerned with the admissible solutions to ARMA–Representations. Again several procedures from the package `linstruct` are used and hence included. The package currently consists of 22 procedures.

The procedures contained within the package `linsol` can be sub–divided into the following areas and several are discussed.

Solution of ARMA–Representations [4] – Procedures for the solution, and corresponding admissibility conditions, of ARMA–Representations described by the matrix equation

$$A(\sigma)y(k) = B(\sigma)u(k) \tag{11}$$

`FORWARD(A,B,q)` – This solves the ARMA–Representation (11) in a *forward* fashion given that a set of admissible initial output conditions exist for a given input sequence. Analogously the procedure `BACKWARD(A,B,q)` solves (11) in a *backward* fashion given that a set of admissible final output conditions

exist for a given input sequence while the procedure `SYMMETRIC(A,B,q,N,k)` solves (11) within the interval $[0,\ N]$ for a combined set of admissible initial and final output conditions.

`FORADMISS(A,B,q,'L')` – Computes a set of admissibility equations for the forward solution of (11) in terms of the initial output conditions and input sequence. Correspondingly solves for a set of admissible initial output conditions if a solution to these equations exists. Analogously the procedures `BACKADMISS(A,B,q,'L')` and `SYMADMISS(A,B,q,N,'L')` compute the set of admissibility equations for the backward and symmetric solutions to (11) respectively.

Various Solution Problems – This includes several procedures connected with the solution of diophantine equations, continuous non–regular AR–Representations and feedback compensation problems [3].

IMPLEMENTATION AND EXAMPLE

An advantage of Maple is that it enables new packages, or groups of related procedures, to be formed and implemented easily. This even includes the option of connecting help files to these packages and their corresponding individual procedures (see [2] for details on this).

In our case we have constructed four packages namely `linstruct`, `linrep` and `linsol`, as detailed above, and the complete package `linsys` which contains ALL the procedures. These are read into each Maple session via the command `>with(<proc>):` where `<proc>` is the required package. For all the packages the existing linear algebra package `linalg` is also required and is read in via `>with(linalg):`.

Below we illustrate how the package `linrep` is implemented in a Maple session and call the procedures `realpoly` and `MINrealpoly` to form a minimal realization of a matrix $A(s) \in \mathbb{R}[s]^{3\times 3}$ as given by

$$A = \begin{pmatrix} s+1 & s^2 & 0 \\ 0 & s+1 & 0 \\ 0 & 0 & 1 \end{pmatrix} \tag{12}$$

These procedures are briefly explained above. The machine used is a SUN SPARC station 10 (75Mhz SuperSPARC II).

```
eros%maple
          libname := /usr/local/maple/lib, /home/Ice/majj2/mymaplelib
```

(The presented packages are all contained within the directory `mymaplelib` and Maple needs to be *pointed* here by the above line to read the individual packages in.)

```
> with(linalg):
Warning: new definition for   norm
Warning: new definition for   trace
> with(linrep):
> A:=matrix(3,3,[s+1,s^2,0,0,s+1,0,0,0,1]);
                                  [          2     ]
                                  [ s + 1    s    0 ]
                                  [                 ]
                            A := [   0    s + 1  0 ]
                                  [                 ]
                                  [   0      0    1 ]
> JBC:=realpoly(A,3,3);
                [ 0  1  0  0  0 ]  [ 0    0   0 ]
                [               ]  [             ]
                [ 0  0  1  0  0 ]  [ 1    0   0 ]  [ 1  0  -1  0   2 ]
                [               ]  [             ]  [                 ]
        JBC := [[ 0  0  0  0  0 ], [ 0    1   0 ], [ 0  1   2  0  -2 ]]
                [               ]  [             ]  [                 ]
                [ 0  0  0  0  0 ]  [ 0    0   1 ]  [ 0  0   0  1   0 ]
                [               ]  [             ]
                [ 0  0  0  0  0 ]  [ 1/2  1/2 0 ]
```

(These are the matrices $J_\infty, B_\infty, C_\infty$ as in (7).)

```
> JBCD:=MINrealpoly(A,3,3);
                [ 0  1  0 ]  [ 0  0  0 ]  [ 1  0  -1 ]  [  1   1  0 ]
                [         ]  [         ]  [           ]  [           ]
      JBCD := [[ 0  0  1 ], [ 1  0  0 ], [ 0  1   2 ], [ -1  -1  0 ]]
                [         ]  [         ]  [           ]  [           ]
                [ 0  0  0 ]  [ 0  1  0 ]  [ 0  0  -0 ]  [  0   0  1 ]
```

(These are the matrices $\hat{J}_\infty, \hat{B}_\infty, \hat{C}_\infty, \hat{D}_\infty$ as in (8).)

```
> quit
bytes used=3862120, alloc=1441528, time=3.62
```

It can be seen that from the above results

$$A(s) = C_\infty(I_5 - sJ_\infty)^{-1}B_\infty = \hat{C}_\infty(I_3 - s\hat{J}_\infty)^{-1}\hat{B}_\infty + \hat{D}_\infty \tag{13}$$

Clearly in this example the realization obtained via the procedure `MINrealpoly` is advantageous as it has lower dimension.

CONCLUSIONS

In this paper we have emphasized the advantages of the symbolic computational language Maple in solving problems in linear systems. We have illustrated this by developing a linear systems package, containing many corresponding procedures, which is easily and efficiency implemented in Maple. It is clear that such a growing package is extremely useful in this area of work where problems tend to be unfeasibly solvable via hand.

References

[1] Bosgra, O. H. and Van Der Weiden, A. J. J. (1981). Realizations in generalized state-space form for polynomial system matrices and the definition of poles, zeros and decoupling zeros at infinity. *Int. Journal of Control* **32** 731–735.

[2] Char, B. W., Geddes, K. G., Gonnet, G. H. and Watt, S. M. (1991). *Maple V Language Reference Manual.* Springer-Verlag.

[3] Jones, J., Karampetakis, N. P. and Pugh, A. C. (1996). Computation of the Generalized Inverse of a Rational Matrix via MAPLE and Applications. *Submitted to CACSD–96, Michigan, U.S.A.*

[4] Karampetakis, N. P., Jones, J. and Antoniou, S. (1996). Forward, Backward And Symmetric Solutions Of Discrete ARMA–Representations. *Submitted to Journal of Circuit Systems & Signal Processing.*

[5] Penrose, R. (1955). A Generalized Inverse for Matrices. *Proc. Cambridge Philos. Soc.* **51** 406–413.

[6] Rosenbrock, H. H. (1974). *State Space and Multivariable Theory.* Nelson, London.

[7] Vardulakis, A. I. G. and Karampetakis, N. P. (1995). On the reduction of a Polynomial Matrix Model of a Linear Multivariable System to Generalized State Space Form. *Submitted to SIAM Journal on Control and Optimization*

[8] Vardulakis, A. I. G. (1991) *Linear Multivariable Control.* Wiley.

[9] Wolfram, S. (1992) *Mathematica.* Addison Wesley, Willow.